W. E. Burr
16 April. 1997

A PKI Interoperability Test Plan

1. Introduction

This test plan describes a series of tests to demonstrate PKI interoperability. For the purposes of this interoperability demonstration test, PKI interoperability exists when two or more *Certification Authorities (CAs)* are linked by certificates or cross-certificates and when the PKI clients for the certificate holders of each of those CAs can validate the digital signatures of certificate holders of the other CAs, by validating a chain of certificates, or *certification path*, from a trusted CA, through other CAs, to the certificate of the signer. That is, we are testing the ability of clients to interoperate with each other, by validating each other's certification paths and signatures, through a PKI.

There are other aspects of CA component interoperability that are not tested here. For example, a client can interoperate with a CA via an automated protocol for certificate issuance, such as that specified in the *Minimum Interoperability Specification for PKI Components (MISPC)* [MISPC 96]. But, clients can interoperate securely with (i.e., validate the certification paths and the signatures of) other clients, if common message formats and application protocols are used, and if suitable certification paths exist, however those certification paths were created, and whether or not the clients use the same certificate issuance protocol.

Moreover, NIST expects to procure a "root CA" which is flexible in its ability to issue certificates and to request them from other CAs. The NIST root CA is primarily intended to certify and be certified by other CAs, and the number of CAs and CA certificates will be small compared to the number of clients client certificates. Given the relatively small number of CAs, the issuance of CA certificates does not necessarily require an automated protocol. Moreover, the certification of CAs should be a deliberate and carefully considered act, and can involve considerable manual intervention, while this would not be tolerable for client certificates, which will be issued in much larger numbers. Therefore, in this test plan, we assume that the root CA can perform the protocols and processes necessary to issue certificates to other CAs and to request and obtain certificates from other CAs.

The processes and protocols required to issue certificates to end-entity clients and to cross-certify pilot CAs with the root CA, or each other, are not without interest; indeed *many valuable practical lessons will probably be learned in doing this, and much of the reason for performing these interoperability tests is to learn those lessons*. But the tests specified in this test plan simply state the needed certificates, and assume that they can be issued. It is important to understand that the CAs, which issue the certificates that establish certification paths, are not further involved in the use of the PKI and the interoperation of clients to validate certification paths and signatures. While the tests cannot be conducted without the certificates, the issuance of the certificates is a precondition for the tests, not a part of the tests Rather it is the clients and, in some cases, also the repository or directory, that are tested, and the tests are passed or failed as a result of the clients ability to find and validate (or not validate) test signatures and certification paths.

In addition, NIST plans to acquire general purpose client software that is capable of processing very general certification paths composed of certificates consistent with the MISPC. We expect to test the ability of this and other clients validate certification paths and signatures.

NIST also plans to acquire a directory that implements the LDAP protocol, to be used as a repository to make certificates and CRLs available to clients. This will be used to demonstrate the ability of clients to find a certification path in an LDAP based repository.

A convincing interoperability demonstration requires clients at different agencies to interact and perform some useful secure interaction through a nontrivial certification path. This allows a demonstration of interoperability between agencies, systems and trust domains. It is also desirable that the clients be from more than one vendor. The tests below are designed so that they require only that clients be capable of sending each other signed messages, and validating the signatures and certification paths that apply to those messages. Thus they could be applied to a variety of client applications. In the near term, it appears that S/MIME clients are likely to be the most practical client application vehicle for testing and demonstrating interoperability. A more detailed explanation of the reasons for this is found in Appendix A.

The philosophy adopted in this demonstration test plan is to begin with a simple test configuration, with two CAs and one type of client, then, in subsequent tests, extend the testing to more complex cases with more CAs, longer certification paths, more complex certification paths with certificate policies and policy mapping, and client implementations from different vendors.

This is an interoperability demonstration test, not a conformance test. There is an unbounded set of possible interoperability tests and this is not intended to be an exhaustive test of any aspect of interoperability.   It is intended to provide reasonable confidence in the ability of clients to use a PKI to interoperate, and to roughly measure the degree to which they do interoperate, in regard to several important PKI interoperability features, but not all such features.

2. PKI Architecture

In the simplest PKI, illustrated in Figure 1(a), there is only a single CA that issues all certificates. This degenerate case is really just a certificate management system, since there is no transfer of trust beyond the domain of the CA and there are no interesting PKI interoperability issues here, however this is a useful test case for showing that the applications that use the PKI can interoperate at the application level.

In a hierarchical PKI, illustrated in Figure 1(b), a single CA called the “root” CA is the foundation of trust for the PKI. The root issues certificates to subordinate CAs, and they in turn may issue certificates to their own subordinate CAs, and so on. Every user validates certification paths back to the public key of the root CA, which is distributed by some authenticated means to every user. There are several advantages to hierarchical PKI topologies:

- the branches of the hierarchy can often be aligned with either organizational structure or policy, or both, simplifying management of the PKI;

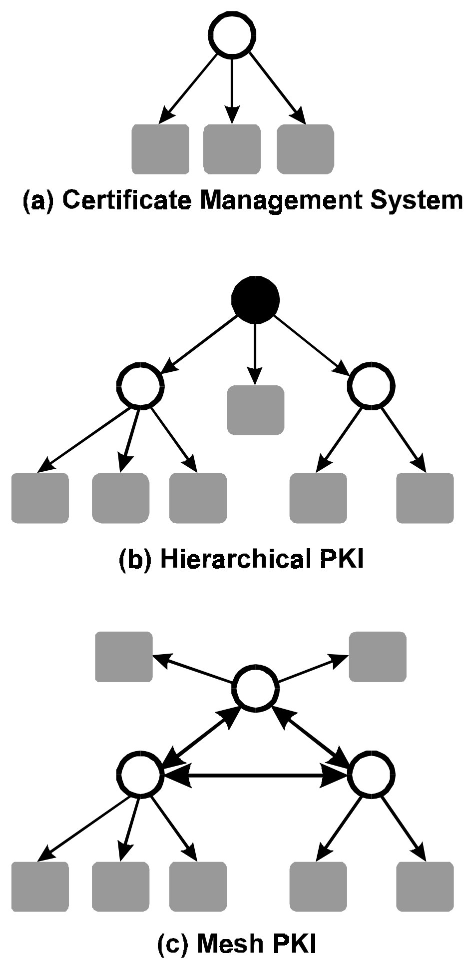

Figure 1 - PKI Topologies

- every certificate holder can build a single certification path from the root to his certificate, and that certification path can be validated by every other certificate holder in the PKI;
- given any end entity certificate, it is comparatively easy to follow the hierarchy back to the root and find the needed certification path.

However, a strictly hierarchical organization is not necessarily congruent with many organizational structures, nor can all desired policies necessarily be expressed hierarchically. If the root CA's key is compromised, the recovery process the entire PKI is shut down and recovery requires authenticated distribution of the new root key to every certificate holder in the entire PKI. Nor is it necessarily easy to agree on a single root for an entire extended nationwide or worldwide PKI.

To go to a more general topology than a hierarchy it is necessary to add a construct, the *cross-certificate.* A cross-certificate is constructed when two CAs each issue a certificate to the other. Thus it expresses a peer relationship, rather than a superior to subordinate relationship. With cross-certificates we can construct a PKI with a mesh topology, as illustrated in Figure 1(c). Although the figure shows a simple PKI with every CA directly connected to every other CA by a cross-certificate, this would not normally be the case for a large PKI. The certification of a CA requires careful consideration, and it would not be practical for every CA to cross-certify with every other CA in a PKI with hundreds or thousands of CAs.

In a mesh PKI, a certificate holder normally bases his trust on the public key of the same CA that issued his certificate. This simplifies the authenticated distribution of the CA's public key and a CA key compromise affects a much smaller part of the PKI than a compromise of a root CA key in a hierarchical PKI. It is also more logical and satisfying for a certificate holder to base his trust on the CA that issued his certificate (which he must in any event trust) than on some remote root CA.

A mesh PKI, however, is more complex than a hierarchical PKI, and organized management of the PKI is more difficult than a hierarchical PKI. It does not necessarily follow that just because CA A cross certifies with CA B, and CA B cross certifies with CA C, that trust always extends from the certificate holders of CA A to the certificate holders of CA C. Therefore the X.509 version 3 certificate includes provision for certificate extensions and defines certain standardized extensions that are useful to manage a PKI. For example, CA A can, by including appropriate extensions in the certificate it issues to CA B, limit or constrain the further propagation of trust to the certificate holders of CA C.

Two of the important extensions for managing PKIs are the Certificate Policies and the Policy Mapping extensions. The Certificate Policies extension allows CAs to identify the policies used to issue certificates in the certificates. The Policy Mapping extension allows a CA to state that one of its particular policies is equivalent to another policy of the CA it issues a certificate to. This interoperability test tests these two standardized extensions.

Finding certification paths in mesh CAs is a more complex problem than in a hierarchical PKI. While algorithms exist to systematically interrogate repositories for certificates until a valid certification path between a trusted CA and any end entity certificate is found (if one exists), it is not clear that such automatic certification path construction facilities will be a part of most commercial clients. Nor would such a process, even if it were implemented, necessarily be quickly resolved. The problem is roughly analogous to finding a path for routing between two nodes in a packet switched network, except that there are many more constraints that may be tested at each node to find a valid certification path.

Further, it should be understood that a mesh topology can always be viewed by a client as a hierarchy, whose root is any CA trusted by the client (of course the client must know the public key of that CA). There are many potential logical hierarchies in any mesh PKI.

However, even in a mesh PKI, it may be useful to designate some CAs in a mesh PKI as "roots" for management or cross certification purposes. In this context a root CA is a well known CA that exists primarily to cross certify with other CAs, particularly other root CAs. It may also imposes a management hierarchy over some part of the PKI. If an organizational CA cross certifies with such a root CA, then it will have some assurance that:

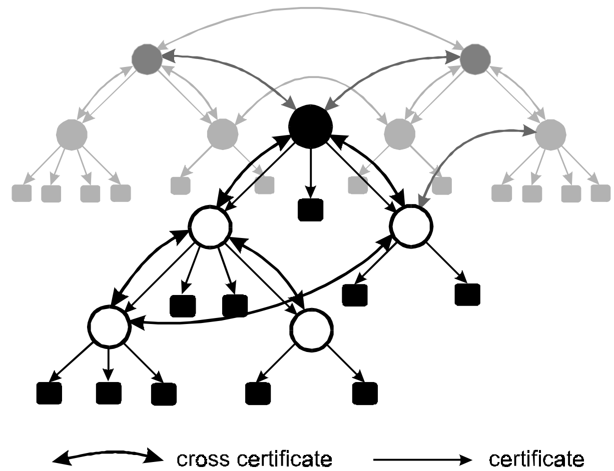

Figure 2 - Hybrid Federal PKI Architecture

- the root is cross certified to many other CAs and all other root CAs, providing relatively efficient (i.e., short) certification paths to other CAs;
- the cross-certificates between the root and other CAs are carefully managed ;
- certificate holders can include a certification path from that well known root CA with signed documents, or otherwise make the path available, in the expectation that certification path will be broadly recognized.

The Federal PKI CONOPS [CONOPS 96] describes a hybrid mesh/hierarchical architecture proposed for the Federal PKI. This architecture is illustrated in Figure 2. In this case there are three conventional hierarchical trees, each under a root CA. The hierarchical CA certificates of each of the trees, however, have parallel cross-certificates, so that CAs also issue certificates to their superiors in the hierarchy. The several root CAs cross certify with each other, and subordinate CAs may also cross-certify each other in non-hierarchical fashion. This allows clients to operate using either their designated root CA as the trusted CA, or using the CA that issued their certificate as the trusted CA. It also allows cross-certificates that implement shorter certification paths than are provided by the hierarchy.

This test plan begins with a few tests that use only a single CA, primarily to test the digital signature validation of the clients in a simple test case. It then moves to progressively more complex topological cases: first two cross-certified CAs, then a root plus two subordinate CA hierarchical PKI, and finally a three CA mesh PKI of cross-certified CAs. In the topologies that follow the root CA is intended to be the NIST root CA that is either used as:

- A root CA in a purely hierarchical PKI, whose public key is the initial starting point for all certification paths, or;
- A designated root CA in mesh PKI. In this case the root's public key is not normally used by clients as the starting point for certification paths, but is primarily a vehicle for broadly cross-certifying other CAs, to facilitate establishment of certification paths;

3. Demonstration Scenarios

While the primary emphasis of the test scenarios below is interactions involving the PKI, when clients from different vendors are to be tested, application level interoperability is also involved and application incompatibilities, rather than certification path processing incompatibilities are a possible reason for failure. Therefore, the first test specified below uses a single CA to ensure that the client application functionality is interoperable, that the digital signature generation and validation processing works, and to ensure that at least a single certificate can be validated correctly.

The scenarios are set up as a general progression from basic to more complex. Since there are many independent, or largely independent, dimensions to PKI interoperability, it is not practical to arrange these tests in order so that each test depends on all the preceding tests, making it sufficient to stop testing, as soon as one test is failed. The tests are arranged from simple to complex to allow confidence in interoperability and knowledge to be built incrementally and systematically, but the fact that a test fails, does not necessarily mean that there is no point to running succeeding tests.

Most of the test cases assume symmetrical clients with a peer-to-peer relationship, and involve exchanging signed messages and validation of the signatures. These tests can be run unmodified with peer to peer type digital signature applications such as S/MIME. However some applications are asymmetrical, that is “client” to “server” oriented. In such cases, the client may sign a message that is validated by a server, or vice versa, but the process inherent in the application may be asymmetrical. The tests below can generally be adopted to such applications by simply omitting the messages and validations from the side that does not perform them.

In the test cases below CAs are identified with letters as CA A, CA B, and so on. Clients (and their certificates) for each CA are then identified by the CA letter and a number. So A1 and A2 are clients with certificates issued by CA A. Where there is only a single client/certificate issued by a CA for a test, then the number is omitted, so client B is the only client used in the test with a certificate issued by CA B.

3.1 Basic Digital Signature Interoperability

This test uses a minimal certificate management system structure, with a single CA, to ensure that clients have interoperable digital signature functionality and are able to correctly evaluate signatures, and validate a single certificate. This is the only test that is intended to test basic digital signature processing interoperability, and basic client functional interoperability, rather than PKI certification path processing functionality. This test should be run before running any of the following tests, using the clients to be tested in subsequent tests to provide assurance that the interoperability failure, if it occurs, is due to certification path processing rather than incompatibilities in the client implementation of the digital signature validation process itself.

Identical client implementations may initially be tested together, to maximize the chance of application level interoperability. However, where more than one implementation of an application is available, they should be tested against each other. Eventually, if several different client implementations are tested, a client application that interoperates successfully with many of the others may be selected as a client reference implementation.

3.1.1 Valid signature and certificates.

Properties Demonstrated

The clients are able to validate the signatures on the messages and the signatures on certificates.

Configuration

Two clients, A1 and A2, are issued valid certificates by a single CA as illustrated in Figure 3. The certificate for Client A1 is supplied to Client A2 with the signed message, or, out of band, as appropriate to the application. Similarly, the certificate for Client A2 is supplied to Client A1 with the signed message, or, out of band, as appropriate to the application.

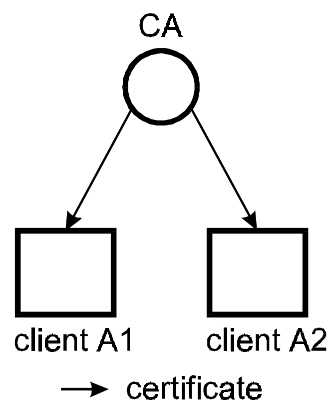

Figure 3 - Basic Interoperability Test Configuration

Test Actions

The two clients exchange signed messages.

Expected Results.

The certification paths and message signatures are validated.

3.1.2 Invalid Signature

Properties Demonstrated

The clients correctly detect signatures that are invalid because a part of the message has been altered.

Configuration

As in test 3.1.1 above.

Test Actions

The two clients generate signed messages. Before the messages are transmitted, however, a character within the signed message envelope is changed. The messages are then exchanged, and the signatures validated.

Expected Results.

The message signatures are found invalid.

3.1.3 Expired certificates

Properties Validated

The clients check expiration dates of certificates and do not validate the signatures on the messages when the certificates have expired.

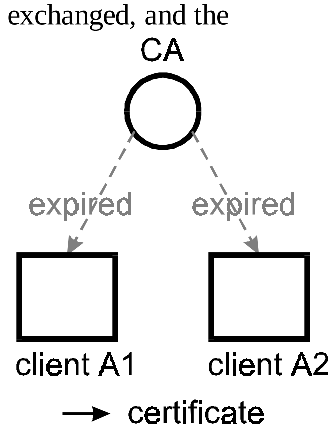

Figure 4 - Expired Certificates Test

Configuration

Two clients, A1 and A2 are issued certificates from a single CA with an expiration date that is earlier than the system time used by the clients, as illustrated in

Figure **4**. The expired certificate for Client A1 is supplied to Client A2 with the signed message, or, out of band, as appropriate to the application. Similarly, the expired certificate for Client A2 is supplied to Client A1 with the signed message, or, out of band, as appropriate to the application.

Test Actions

The two clients exchange signed messages.

Expected Results.

The message signatures are not validated.

3.2 Simple PKI Certification Path Interoperability

These tests demonstrate PKI certification path validation functionality between two clients, A and B, each with certificates from a different CA, when these CAs are cross certified, to make a simple PKI. All tests in this section are predicated upon the successful completion of the tests specified in 2.1 above.

In these tests the NIST root CA may take the role of CA A, or two pilot CAs may perform the tests directly, since the tests are symmetrical and do not distinguish a particular role for a root CA.

3.2.1 Basic Interoperability

This test demonstrates the most fundamental PKI certificate path processing, and unless this test is passed no further tests can be run. It is predicated only on successful completion by clients of the tests under 3.1 above.

Properties Demonstrated

This demonstration shows that the clients can use certificates issued by different cross-certified CAs, to validate a two step certification path.

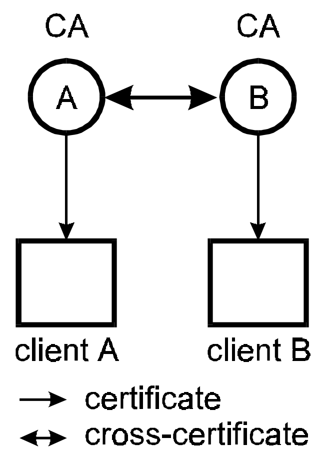

Figure 5 - Basic 2 CA Configuration

Demonstration Configuration

The test certification path topology is shown in Figure 5. In this test, CA A cross certifies with CA B. Each CA issues a certificate to a client. Needed certification paths are provided to the clients, either with the signed messages, or by out of band means as appropriate to the applications.

Demonstration Actions

The two clients exchange signed messages and validate the signatures on the messages. Necessary certification paths are provided with the signed transactions, or given to the clients out of band.

Expected Results

Both clients validate the signatures on the message from the other client.

3.2.2 Basic Certificate Policies Processing,

This test is predicated on successful completion of test 3.2.1 above. This test is applicable only for clients that process the certificate policies extension.

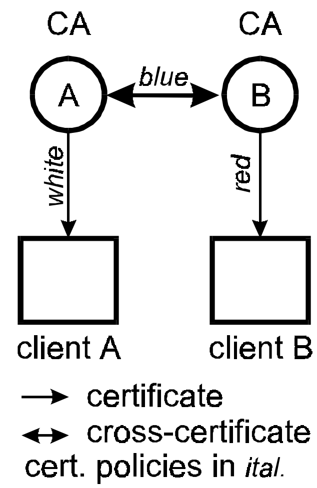

Figure 6 - Basic Policies Configuration

Properties Demonstrated

This demonstration shows that the clients can process certificate policies when validating certification paths.

Demonstration Configuration

The test certification path topology is shown in Figure 6. Client A is issued a certificate by CA A and client B is issued a certificate by CA B. Client B is issued a certificate with a Certificate Policies value of *red*. Client A is issued a certificate with a Certificate Policies value of *white*. Each of the certificates issued to the CAs has a Certificate Policies value of *blue*. The Certificate Policies extension is flagged "noncritical." Needed certification paths are provided to the clients, either with the signed messages, or by out of band means as appropriate to the applications.

A) Valid Policy Sets

Demonstration Actions

The *initial-policy-set* (a list of one or more certificate policy identifiers, indicating that any one of these policies would be acceptable to the certificate user for the purpose of certification path processing) for Client A is set to *blue* and *red*. The *initial-policy-set* for Client B is set to *blue* and *white.* The two clients exchange signed messages and validate the signatures on the messages. Necessary certification paths are provided with the signed transactions, or given to the clients out of band.

Expected Results

Both clients validate the signatures on the message from the other client.

B) Invalid Policy Sets

Demonstration Actions

The *initial-policy-set* for Client A is set to *blue* and *white*. The *initial-policy-set* for Client B is set to *white*. The two clients exchange signed messages and validate the signatures on the messages. Necessary certification paths are provided with the signed transactions, or given to the clients out of band.

Expected Results

Both clients do not validate the signatures on the message from the other client, since each certification path contains a step with no valid policy.

3.2.3 Policy Mapping Processing,

This test is predicated on successful completion of test 3.2.2 above. This test is applicable only for clients that process the certificate policies extension and policy mapping.

Properties Demonstrated

This demonstration shows that the clients can process mapped certificate policies when validating certification paths.

Demonstration Configuration

The test certification path topology is shown in Figure 7. Client A is issued a certificate by CA A with a certificate Policies value of *silver* and client B is issued a certificate by CA B with a Certificate Policies value of *red*. In the cross-certificate between CA A and CA B, the certificate issued by CA A to CA B has a Certificate Policies value of *silver* with a Certificate Mapping that maps *red* to *silver*. The Certificate Policies extension is flagged “noncritical.” The certificate issued by CA B to CA A has a Certificate Policies” value of *red* and a certificate mapping that maps *red* to *silver*. Needed certification paths are provided to the clients, either with the signed messages, or by out of band means as appropriate to the applications.

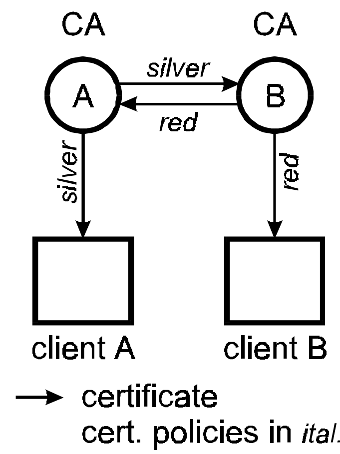

Figure 7 - Policy Mapping Configuration

Demonstration Actions

The *initial-policy-set* for Client A is set to *red*. The *initial-policy-set* for Client B is set to *silver*. The two clients exchange signed messages and validate the signatures on the messages. Necessary certification paths are provided with the signed transactions, or given to the clients out of band.

Expected Results

Both clients validate the signatures on the message from the other client.

3.2.4 Repository Processing

This test demonstrates the clients can obtain the needed certificates from a repository. It is predicated only on successful completion by clients of the test 3.2.1 above.

Properties Demonstrated

This demonstration shows that the clients can use an LDAP repository to find the certificates it needs to validate a two step certification path.

Demonstration Configuration

The test certification path topology is shown in Figure 5. In this test, the certificates used in test step 3.2.1 above are used, however the various certificates are stored in an LDAP repository under the names of the certificate holders, rather than provided directly to the clients. The clients are set to find certificates in the repository.

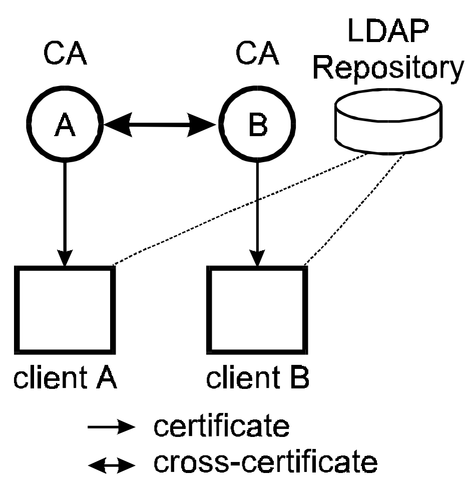

Figure 8 - Basic 2 CA Configuration

Demonstration Actions

The two clients exchange signed messages and validate the signatures on the messages. Necessary certification paths are found by the clients through the repository.

Expected Results

Both clients validate the signatures on the message from the other client.

3.2.5 CRL Interoperability

This test is predicated on successful completion by clients of the test given in 3.2.1 above.

Properties Demonstrated

This test demonstrates that the clients can process a Certificate Revocation List (CRL).

Demonstration Configuration

The test certification path topology is shown in Figure 5. The certificates defined in step 3.2.1 above are used, however each CA also creates a CRL that revoke the two client certificates.

Demonstration Actions

The two clients exchange signed messages and validate the signatures on the messages. Necessary certification paths and CRLs are provided with the signed transactions, or given to the clients out of band.

Expected Results

Both clients reject the signatures on the message from the other client because of an invalid certification path.

3.2.6 CRL Interoperability with Directories

This test is predicated on successful completion by clients of the tests given in 3.2.1 and 3.2.5 above.

Properties Demonstrated

This test demonstrates that the clients can retrieve and process Certificates and a Certificate Revocation List (CRL) from an LDAP repository.

Demonstration Configuration

The test certification path topology is shown in Figure 8. The certificates and CRLs defined in step 3.2.5 above are used.

Demonstration Actions

The two clients exchange signed messages and validate the signatures on the messages. Certification paths and CRLs are not provided with the signed transactions, or given to the clients out of band. The client is given the address of the repository and set to retrieve certificates and CRLs from that repository.

Expected Results

Both clients reject the signatures on the message from the other client because of an invalid certification path.

3.3 Root CA Hierarchical PKI Interoperability Tests.

These tests involve the root CA and somewhat more complex topology. In these test the root CA is the CA trusted by all clients. It may not be possible to initialize every client to use the root CA, that does not issue the client his certificate as the trusted CA, but most clients should allow this.

3.3.1 Basic Hierarchical Interoperability

This test shows interoperability of clients in a hierarchical PKI.

Properties Demonstrated

The ability of clients to correctly process certification paths in a hierarchical PKI where the root CA is the source of all certification paths.

Demonstration Configuration

The certification path topology is illustrated in Figure 9. Pilot CAs A and B are issued certificates from the root with a Certificate Policies value of *blue* CA A issues client A certificate with a Certificate Policies value of *white*, while CA B issues client B a certificate with a certificate policies value of *red*. Clients A and B are set to trust the Root CA and it's public key. Needed certification paths are provided to the clients, either with the signed messages, or by out of band means as appropriate to the applications.

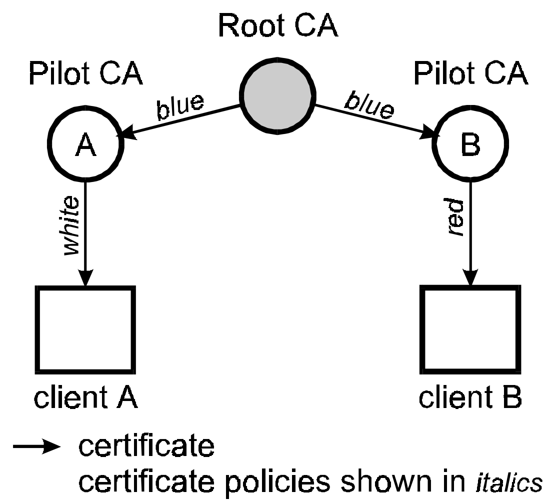

Figure 9 - Basic Hierarchical PKI

A) No Policies

Demonstration Actions

Clients A and B are set ignore Certificate Policies or to accept any policy. They exchange signed messages.

Expected Results

The clients validate each other's signed messages.

B) Valid Policies

Demonstration Actions

The *initial-policy-set* for Client A is set to *blue* and *red*. The *initial-policy-set* for client B is set to *blue* and *white* The clients exchange signed messages.

Expected Results

The client's validate each other's signed messages.

C) Invalid Policies

Demonstration Actions

The *initial-policy-set* for Client A is set to *blue* and *white*." *initial-policy-set* for client B is set to *red* and *white*. The clients exchange signed messages.

Expected Results

Both clients do not validate the certification path due to invalid policies.

3.3.2 Hierarchical Interoperability with Directories

This test shows interoperability of clients in a hierarchical PKI to find certification paths using a repository.

Properties Demonstrated

The ability of clients to find certification paths in directories and correctly process them in a hierarchical PKI where the root CA is the source of all certification paths.

Demonstration Configuration

The certification path topology is illustrated in Figure 10. Pilot CAs A and B are issued certificates from the root with a Certificate Policies value of *blue*. CA A issues client A certificate with a Certificate Policies value of *white*, while CA B issues client B a certificate with a certificate policies value of *red*. Clients A and B are set to trust the Root CA and it's public key. Needed certification paths are provided to the clients, either with the signed messages, or by out of band means as appropriate to the applications.

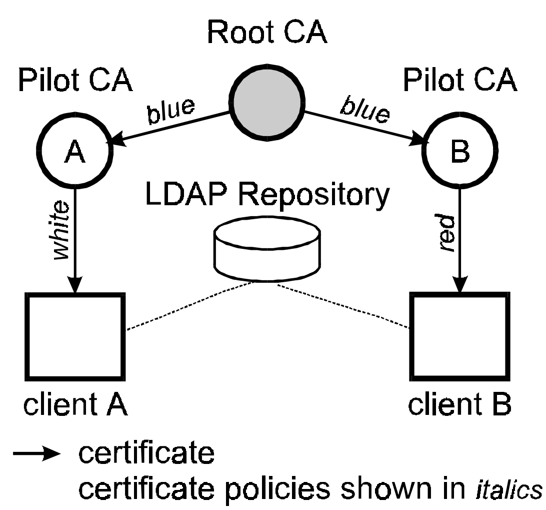

Figure 10 - Hierarchical PKI and Repository

A) No Policies

Demonstration Actions

Clients A and B are set to ignore certificate policies or to accept any policy. They exchange signed messages.

Expected Results

The clients validate each other's signed messages.

B) Valid Policies

Demonstration Actions

The *initial-policy-set* for Client A is sets to *blue* and *red*. The *initial-policy-set* for Client B is set to *blue* and *white*." The clients exchange signed messages.

Expected Results

The clients validate each other's signed messages.

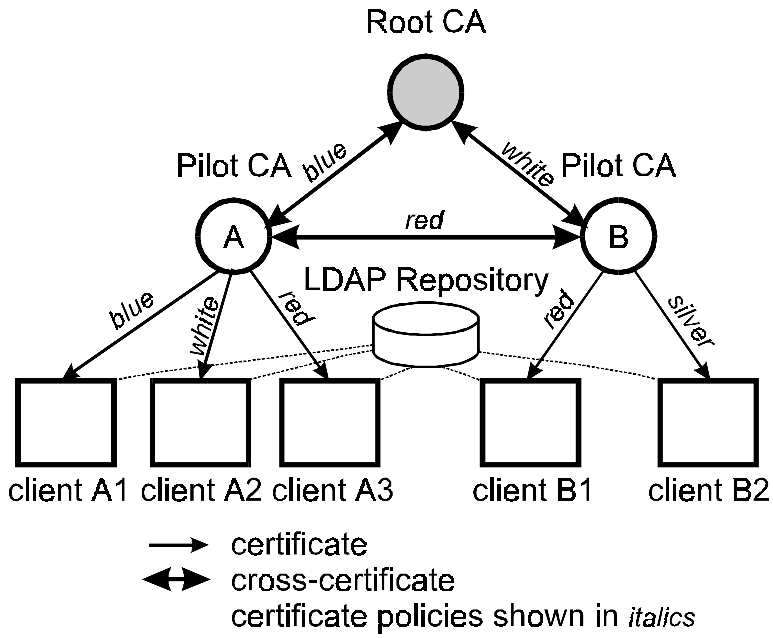

Figure 11 - Mesh PKI and Repository

C) Invalid Policies

Demonstration Actions

The *initial-policy-set* for client A is set to *blue* and *white*. The *initial-policy-set* for Client B is set too *red* and *white*. The clients exchange signed messages.

Expected Results

Both clients do not validate the certification path due to invalid policies.

3.4 Mesh Topology PKI Interoperability Tests.

These tests involve the root CA and general mesh topology. In these test either the root CA or a pilot CA may be the origin of certification paths for different clients, and there are alternative trust paths for different clients.

The certification path topology for all tests in this section is illustrated in Figure 11. Two Pilot application CAs, A and B, are cross certified with the NIST root CA. CA A and CA B also directly cross-certify with each other. CA A issues certificates to three clients A1, A2 and A3. CA B issues certificates to two clients B1 and B2. It is not a requirement of this demonstration test that different client implementations be used.

Certificates are issued as shown in the table below:

Certificate	**Certificate Policy Values**	**Policy Mappings**
Issued by root to CA A	*red, white, blue*	none
Issued by root to CA B	*blue*	*white* maps to *silver*
Issued by CA A to root	*red, white, blue*	none
Issued by CA A to client A1	*blue*	none
Issued by CA A to client A2	*white*	none
Issued by CA A to client A3	*red*	none
Issued by CA A to CA B	*red*	*white* maps to *silver*
Issued by CA B to root	*white*	*silver* maps to *white*
Issued by CA B to CA A	*red*	none
Issued by CA B to Client B1	*red*	none
Issued by CA B to client B2	*silver*	none

The repository is configured to include the certificates listed above.

3.4.1 Policies Not used.

Properties Demonstrated

This test demonstrates the ability to use the repository to find a certification path and to validate it without certificate policies. Note that there are two possible paths.

Demonstration Configuration

Client A1 is set to accept all certificate policies. Client B1 sends a signed message to client A1.

Expected Results

Client A1 validates client B1's signature.

3.4.2 Certificate Policies

Properties Demonstrated

This test demonstrates the ability to use the repository to find a certification path and to validate it correctly using certificate policies. Note that there are two possible paths between all A and B certificates

Demonstration Configuration

The *initial-policy-set* for Client A1, A1 and A3 are set as shown below:

Client	***initial-policy-set***
A1	*blue* and *white*
A2	*blue, white* and *red*
A3	*red*:

Client B1 sends a signed message to clients A1, A2 and A3 Certification paths are not provided with the signed messages..

Expected Results

Expected Results are shown below:

Client	**Results**
A1	Signature not validated, since there is no path with only *blue* or *white* Certificate Policies values
A2	Signature validated
A3	Signature validated

3.4.3 Policy Mapping

Properties Demonstrated

This test demonstrates the ability to use the repository to find a certification and to validate it correctly using certificate policies and policy mapping. Note that there are two possible paths between all A and B certificates

Demonstration Configuration

The *initial-policy-set* for Client A1, A1 and A3 are set as shown below:

Client	***initial-policy-set***
A1	*blue* and *white*
A2	*blue, white* and *red*
A3	*red*

Client B2 sends a signed message to clients A1, A2 and A3. Certification paths are not provided with the signed messages.

Expected Results

Expected Results are shown below:

Client	**Results**
A1	Signature validated,
A2	Signature validated
A3	Signature not validated since *silver* maps to *white*

3.4.4 CRL processing

Properties Demonstrated

This test demonstrates the ability to use the repository to find a CRL and certification path and to validate it correctly using certificate policies and CRLs.

Demonstration Configuration

A CRL is added to the repository showing the certificate issued by CA A to CA B as revoked. The *initial-policy-set* for Client A1, A1 and A3 are set as shown below:

Client	***initial-policy-set***
A1	*blue* and *white*
A2	*blue, white* and *red*
A3	*red*

Client B1 sends a signed message to clients A1, A2 and A3 Certification paths are not provided with the signed messages..

Expected Results

Expected Results are shown below:

Client	**Results**
A1	Signature not validated, since there is no path with only *blue* or *white* Certificate Policies values
A2	Signature validated
A3	Signature not validated, since only *red* certification path contains a revoked certificate

4. Test Summary

The illustrations and tables shown below illustrate the functionality tested in each test case.

4.1 Digital Signature

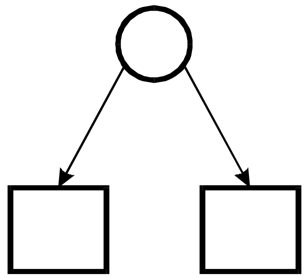

Figure 12 - Certification Path Topology

Table 1 - Digital Signature Tests

Function Tested	**Test Case** 3.1.1	3.1.2	0
Validate Signature	✓		
Detect Invalid Signature		✓	
Detect Expired Certificates			✓

4.2 Simple PKI

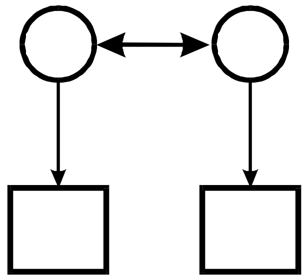

Figure 13 - Certification Path Topology

Table 2 - Simple PKI

	Test Case					
Function Tested	3.2.1	3.2.2 A	3.2.2 B	3.2.3	3.2.4	3.2.5
Signature Chain Processing	✓	✓		✓	✓	✓
Valid Certificate Policies		✓		✓		
Invalid Certificate Policies			✓			
Certificate Policy Mapping				✓		
Repository					✓	
CRL Processing						✓

4.3 Hierarchical PKI

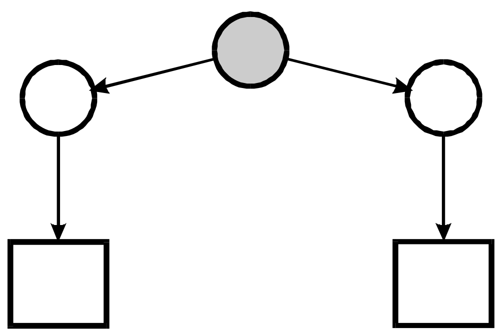

Figure 14 - Hierarchical Certification Path Topology

Table 3 - Hierarchical PKI

Function Tested	3.3.1 A	3.3.1 B	3.3.1 C	3.3.2 A	3.3.2 B	3.3.2 C
Certification paths provided to clients						
Signature Chain Processing	✓	✓				
Valid Certificate Policies		✓				
Invalid Certificate Policies			✓			
Repository *used to find Certification Paths*						
Signature Chain Processing				✓	✓	
Valid Certificate Policies					✓	
Invalid Certificate Policies						✓

4.4 Mesh PKI

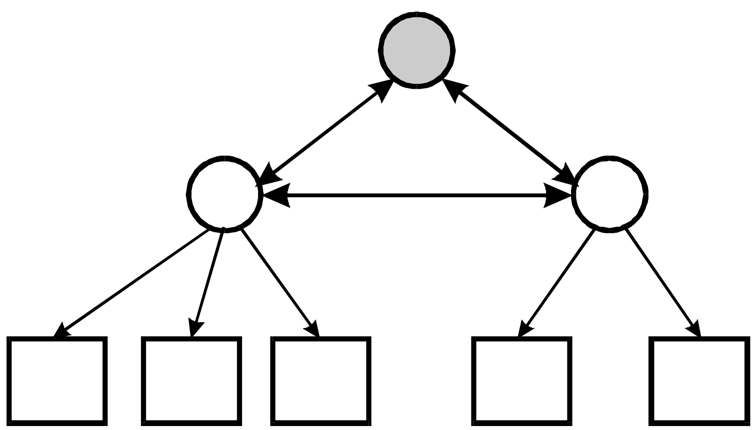

Figure 15 - Mesh Certification Path Topology

Table 4 - Mesh PKI

Function Tested	Test Case 3.4.1	3.4.2	3.4.3	3.4.4
Signature Chain Processing	✓	✓	✓	✓
Find Cert. Path in Repository	✓	✓	✓	✓
Valid Certificate Policies		✓	✓	✓
Invalid Certificate Policies		✓	✓	✓
Policy Mapping			✓	✓
CRL Processing				✓

5. References

[CONOPS 96] *Public Key Infrastructure Technical Specification: Part C - Concept of Operations*, William E. Burr. Available from http://csrc.nist.gov/pki

[MISPC 96] William Burr. Donna Dodson, Noel Nazario, and W. Timothy Polk**,** *Minimum Interoperability Specification for PKI Components,* NIST,Dec. 2, 1996. Available from http://csrc.nist.gov/pki

[RFC 1521] N. Borenstein, N. Freed, RFC 1521: MIME (Multipurpose Internet Mail Extensions) Part One: Mechanism for Specifying and Describing the Format of Internet Message Bodies, Sept. 1993.

[PKCS #7] RSA laboratories, PKCS #7: Cryptographic Message Syntax Standard, Version 1.5, November 1993.

[PKCS #10] RSA Laboratories, PKCS #10: Certification Request Syntax Standard, Version 1.0, November 1993.

[RFC 1847] J Galvin, S Murphy, S. Crocker, N.Freed, RFC 1847: Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted, October 1955.

Appendix A: Choice of the client application

This proposed demonstration requires compatible clients, ideally from different vendors, in several agencies. What client is likely to be available and practical for such a demonstration? S/MIME clients appear to be the best choice for the interoperability demonstration.

S/MIME stands for Secure/Multipurpose Internet Mail Extensions [RFC 1847]. The S/MIME specification is being developed by RSA Laboratories, in cooperation with other vendors and has been submitted to the Internet Engineering Task Force. It provides a mechanism for signing and encrypting MIME e-mail attachments. MIME (Multipurpose Internet Mail Extensions, [RFC 1521]) has become the accepted standard for attaching various binary files that require special encoding to be processed by many SMTP (Simple Mail transport Protocol [RFC 822]) servers.

RSA Labs plans to propose S/MIME as an Internet standard, however at present it is under the control of RSA and its industry partners. Moreover it uses several proprietary RSA "standards" ([PKCS #7] and [PKCS #10]). Therefore S/MIME has a somewhat proprietary flavor, which is a disadvantage. However, applications standards developed by a more open consensus process, that make use of a PKI, are not as widely implemented.

The primary reason for using S/MIME for this demonstration is its wide acceptance and implementation. Major vendors endorsing the S/MIME secure interoperable e-mail plan include: Microsoft, Lotus, Banyan, VeriSign, ConnectSoft, QUALCOMM, Frontier Technologies, Network Computing Devices, FTP Software, Wollongong, SecureWare and RSA Data Security. Some vendors, including Deming Software, Frontier Technology, Netscape, Nortel and OpenSoft already have S/MIME implementations and are presently participating in interoperability tests.

Therefore interoperable S/MIME implementations from multiple vendors seems assured. Moreover, the secure messaging implemented by S/MIME will undoubtedly be the basic security foundation upon which many higher level Federal PKI applications will be built. It is likely that future Federal PKI applications, such as purchasing, travel, timekeeping and the like may all rely on S/MIME secure messaging. In addition, Federal users will use S/MIME clients directly to send signed or encrypted e-mail.

For this interoperability test we expect the client will be able to sign S/MIME messages and validate the signatures and certification paths for signed S/MIME messages. Encryption, also provided by S/MIME, is irrelevant to this interoperability specification.

The purpose of this test is to the ability of the pilot project and root CAs to issue the needed certificates, and of the client certification path processing functions to find and validate certification paths. This will not be an S/MIME test, per se; the only S/MIME functionality required is to be able to sign S/MIME encoded messages and validate the signatures.